Seventh Workshop on the DOE ACTS Collection

Gearing up Scientific Applications for the Petascale Computing Era

August 22-25, 2006

Osni Marques
Lawrence Berkeley National Laboratory
OAMarques@lbl.gov

Outline

- Keeping the pace with the software and hardware
  - Hardware evolution
  - Performance tuning
  - Software selection
  - Other challenges
- The DOE ACTS Collection Project
  - Goals
  - Current features
  - A sample of scientific applications
  - Outreach efforts

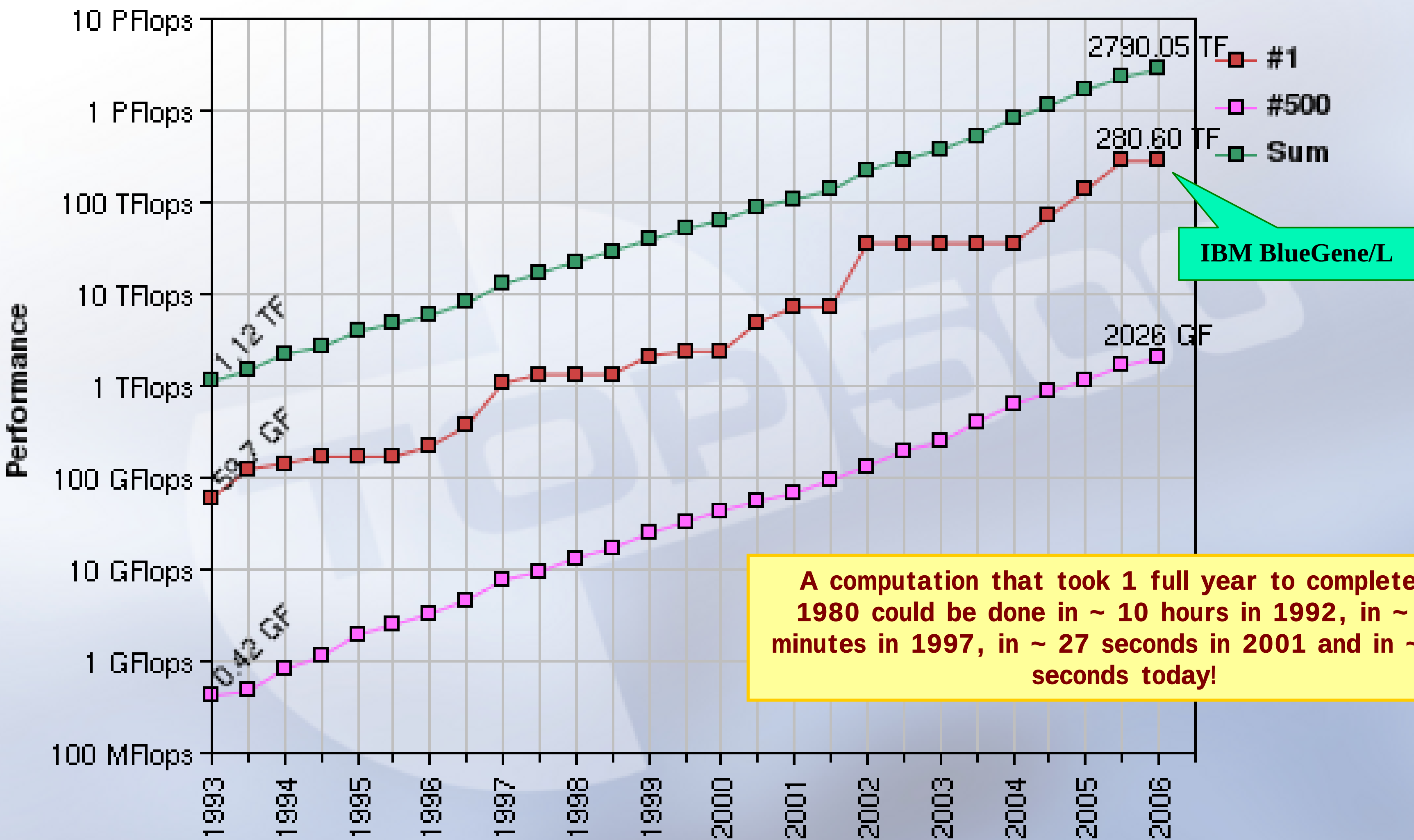
Challenges in the Development of Scientific Codes

Scientific Computing: *Third Pillar of Science*

- Research in computational sciences is fundamentally interdisciplinary
- The development of complex simulation codes on high-end computers is not a trivial task
- Productivity
  - Time to the first solution (prototype)
  - Time to solution (production)
  - Other requirements
- Complexity
  - Increasingly sophisticated models
  - Model coupling
  - Interdisciplinarity
- Performance
  - Increasingly complex algorithms
  - Increasingly complex architectures
  - Increasingly demanding applications

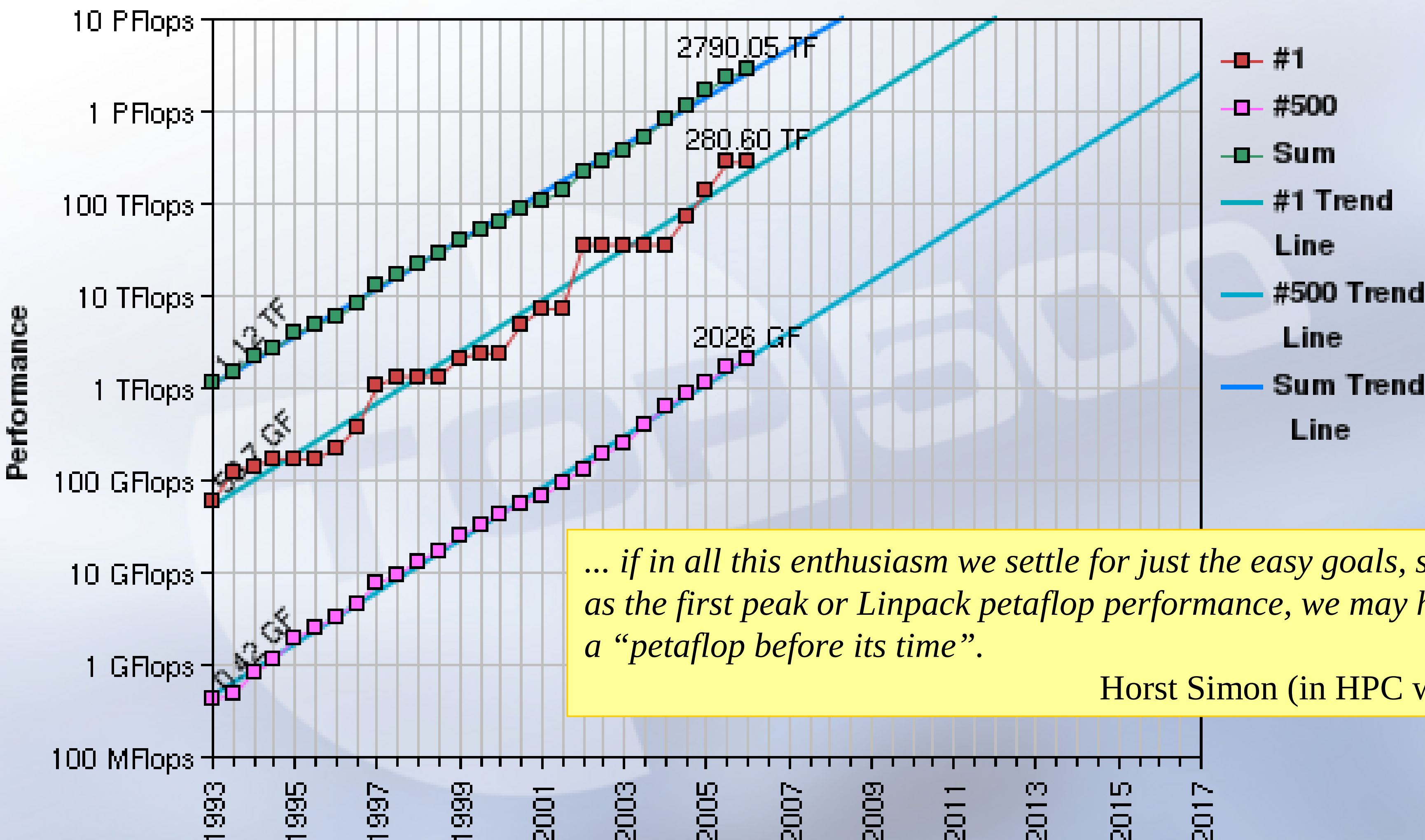
- Libraries written in different languages
- Discussions about standardizing interfaces are often sidetracked into implementation issues
- Difficulties managing multiple libraries developed by third-parties
- Need to use more than one language in one application
- The code is long-lived and different pieces evolve at different rates
- Swapping competing implementations of the same idea and testing without modifying the code
- Need to compose an application with some other(s) that were not originally designed to be combined

Performance Development



A computation that took 1 full year to complete in 1980 could be done in ~ 10 hours in 1992, in ~ 16 minutes in 1997, in ~ 27 seconds in 2001 and in ~ 1.7 seconds today!

Projected Performance Development



... if in all this enthusiasm we settle for just the easy goals, such as the first peak or Linpack petaflop performance, we may have a "petaflop before its time".

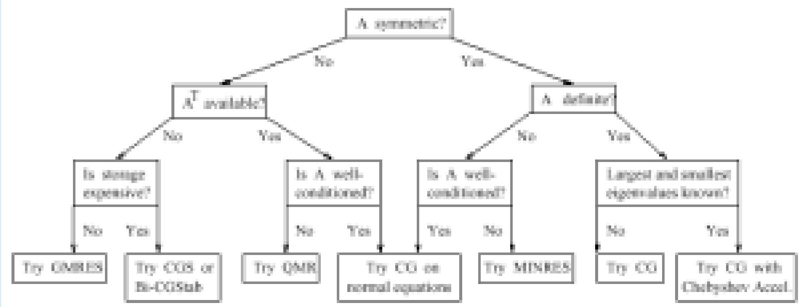
Horst Simon (in HPC wire)

What About Software Selection?

Example: $Ax = b$

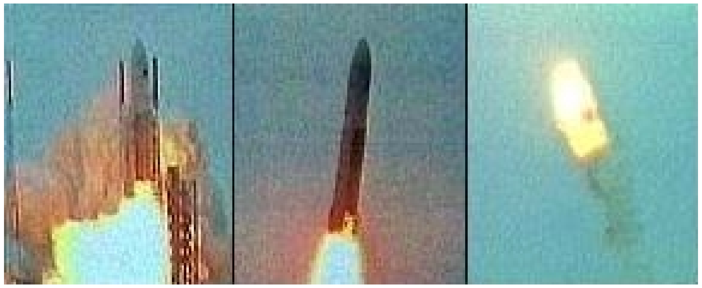
- Use a direct solver (A=LU) if
  - Time and storage space acceptable
  - Iterative methods don't converge
  - Many b's for same A
- Criteria for choosing a direct solver
  - Symmetric positive definite (SPD)
  - Symmetric
  - Symmetric-pattern
  - Unsymmetric
- Row/column ordering schemes available
  - MMD, AMD, ND, graph partitioning
- Hardware

Build a preconditioning matrix K such that Kx=b is much easier to solve than Ax=b and K is somehow "close" to A (incomplete LU decompositions, sparse approximate inverses, polynomial preconditioners, preconditioning by blocks or domains, element-by-element, etc). See *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods.*



Bugs…

On February 25, 1991, during the Gulf War, an American Patriot Missile battery in Dharan, Saudi Arabia, failed to track and intercept an incoming Iraqi Scud missile. The Scud struck an American Army barracks, killing 28 soldiers and injuring around 100 other people. The problem was an inaccurate calculation of the time since boot due to computer arithmetic errors.



On June 4, 1996, an Ariane 5 rocket launched by the European Space Agency exploded just forty seconds after its lift-off from Kourou, French Guiana. The rocket was on its first voyage, after a decade of development costing $7 billion. The problem was a software error in the inertial reference system. Specifically a 64 bit floating point number relating to the horizontal velocity of the rocket with respect to the platform was converted to a 16 bit signed integer.

On August 23,1991, he first concrete base structure for the Sleipner A platform sprang a leak and sank under a controlled ballasting operation during preparation for deck mating in Gandsfjorden outside Stavanger, Norway. The post accident investigation traced the error to inaccurate finite element approximation of the linear elastic model of the tricell (using the popular finite element program NASTRAN). The shear stresses were underestimated by 47% leading to insufficient design. In particular, certain concrete walls were not thick enough.

http://wwwzenger.informatik.tu-muenchen.de/persons/huckle/bugse.html

What is the ACTS Collection?

http://acts.nersc.gov

- **A**dvanced **C**ompu**T**ational **S**oftware Collection
- Tools for developing parallel applications
- ACTS started as an “umbrella” project

Goals

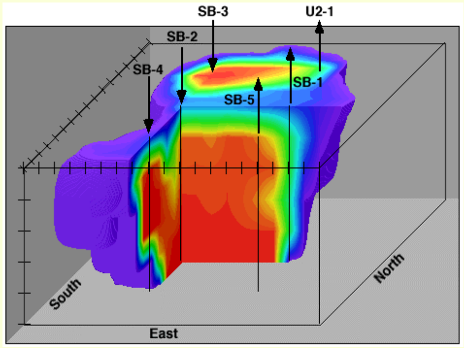
- *Extended support for experimental software*
- *Make ACTS tools available on DOE computers*
- *Provide technical support (acts-support@nersc.gov)*
- *Maintain ACTS information center (http://acts.nersc.gov)*
- *Coordinate efforts with other supercomputing centers*
- *Enable large scale scientific applications*
- *Educate and train*

- ***High***
  - Intermediate level
  - Tool expertise
  - Conduct tutorials
  - ***Intermediate***
    - Basic level
    - Higher level of support to users of the tool
    - ***Basic***
      - Help with installation
      - Basic knowledge of the tools
      - Compilation of user’s reports

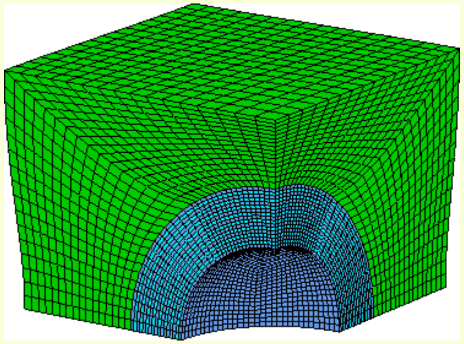
Current ACTS Tools and their Functionalities

Category	Tool	Functionalities
Numerical	**Aztec**	Algorithms for the iterative solution of large sparse linear systems.
	Hypre	Algorithms for the iterative solution of large sparse linear systems, intuitive grid-centric interfaces, and dynamic configuration of parameters.
	PETSc	Tools for the solution of PDEs that require solving large-scale, sparse linear and nonlinear systems of equations.
	OPT++	Object-oriented nonlinear optimization package.
	SUNDIALS	Solvers for the solution of systems of ordinary differential equations, nonlinear algebraic equations, and differential-algebraic equations.
	ScaLAPACK	Library of high performance dense linear algebra routines for distributed-memory message-passing.
	SuperLU	General-purpose library for the direct solution of large, sparse, nonsymmetric systems of linear equations.
	TAO	Large-scale optimization software, including nonlinear least squares, unconstrained minimization, bound constrained optimization, and general nonlinear optimization.
Code Development	**Global Arrays**	Library for writing parallel programs that use large arrays distributed across processing nodes and that offers a shared-memory view of distributed arrays.
	Overture	Object-Oriented tools for solving computational fluid dynamics and combustion problems in complex geometries.
Code Execution	**CUMULVS**	Framework that enables programmers to incorporate fault-tolerance, interactive visualization and computational steering into existing parallel programs.
	TAU	Set of tools for analyzing the performance of C, C++, Fortran and Java programs.
Library Development	**ATLAS**	Tools for the automatic generation of optimized numerical software for modern computer architectures and compilers.

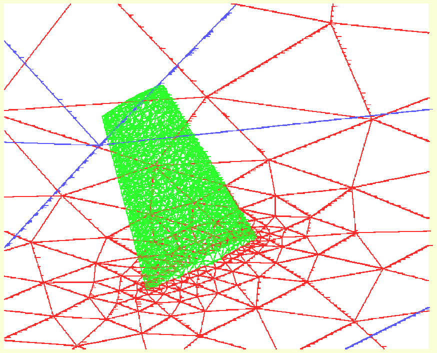
Use of ACTS Tools



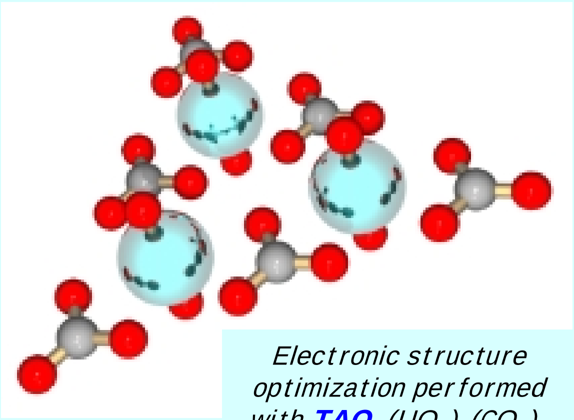
*Multiphase flow using **PETSc**, 4 million cell blocks, 32 million DOF, over 10.6 Gflops on an IBM SP (128 nodes), entire simulation runs in less than 30 minutes (Pope, Gropp, Morgan, Seperhrnoori, Smith and Wheeler).*



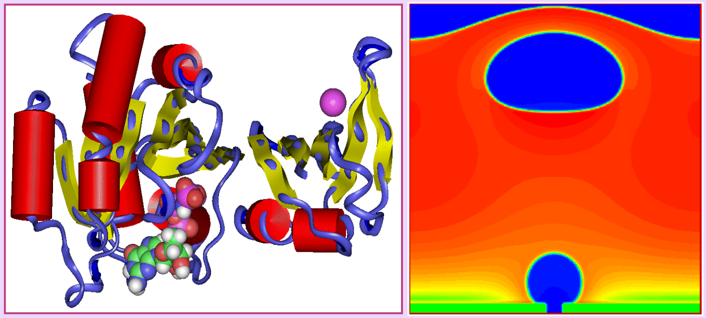
*Model of a "hard" sphere included in a "soft" material, 26 million d.o.f. Unstructured meshes in solid mechanics using Prometheus and **PETSc** (Adams and Demmel).*



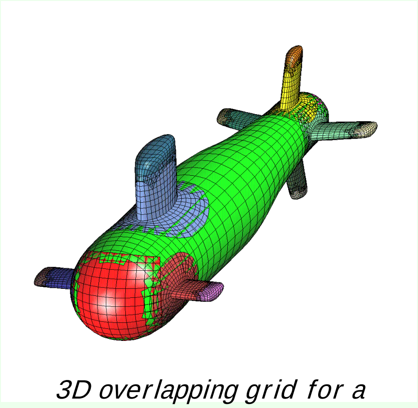
*3D incompressible Euler, tetrahedral grid, up to 11 million unknowns, based on a legacy NASA code, FUN3d (W. K. Anderson), fully implicit steady-state, parallelized with **PETSc** (courtesy of Kaushik and Keyes).*



*Electronic structure optimization performed with **TAO**, $(UO_2)_3(CO_3)_6$ (courtesy of deJong).*

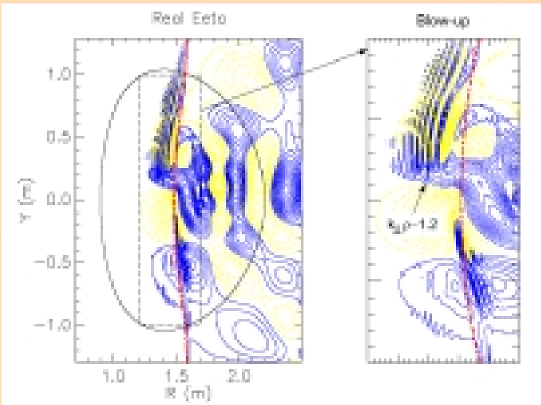


*Molecular dynamics and thermal flow simulation using codes based on **Global Arrays**. GA have been employed in large simulation codes such as NWChem, GAMESS-UK, Columbus, Molpro, Molcas, MWPhys/Grid, etc.*

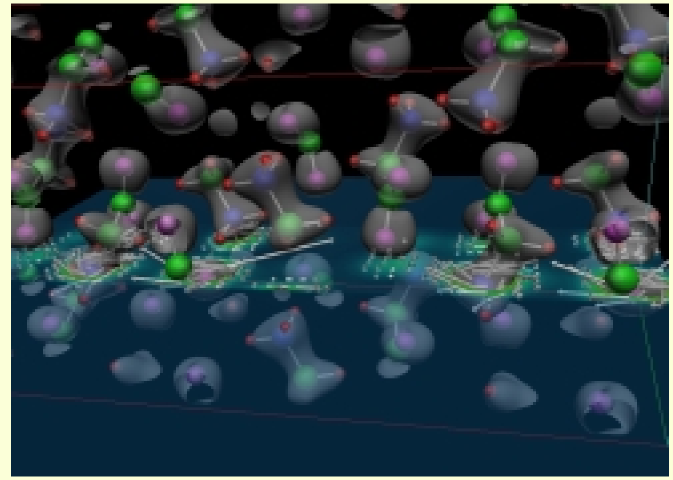


*3D overlapping grid for a submarine produced with **Overture**'s module ogen.*

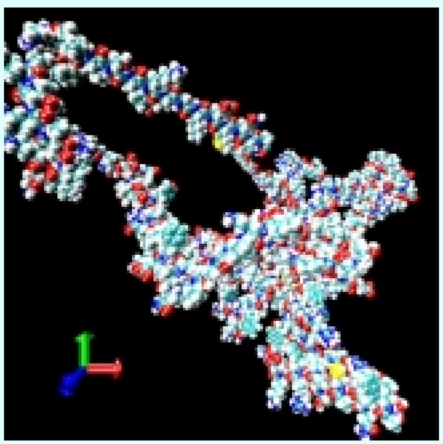
Use of ACTS Tools



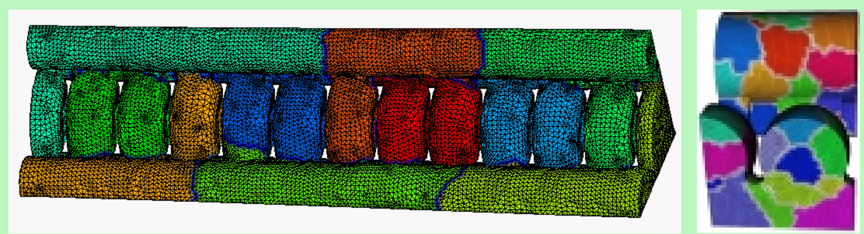
Two ***ScaLAPACK*** *routines, PZGETRF and PZGETRS, are used for solution of linear systems in the spectral algorithms based AORSA code (Batchelor et al.), which is intended for the study of electromagnetic wave-plasma interactions. The code reaches 68% of peak performance on 1936 processors of an IBM SP.*



Induced current (white arrows) and charge density (colored plane and gray surface) in crystallized glycine due to an external field (Louie, Yoon, Pfrommer and Canning), eigenvalue problems solved with ***ScaLAPACK****.*

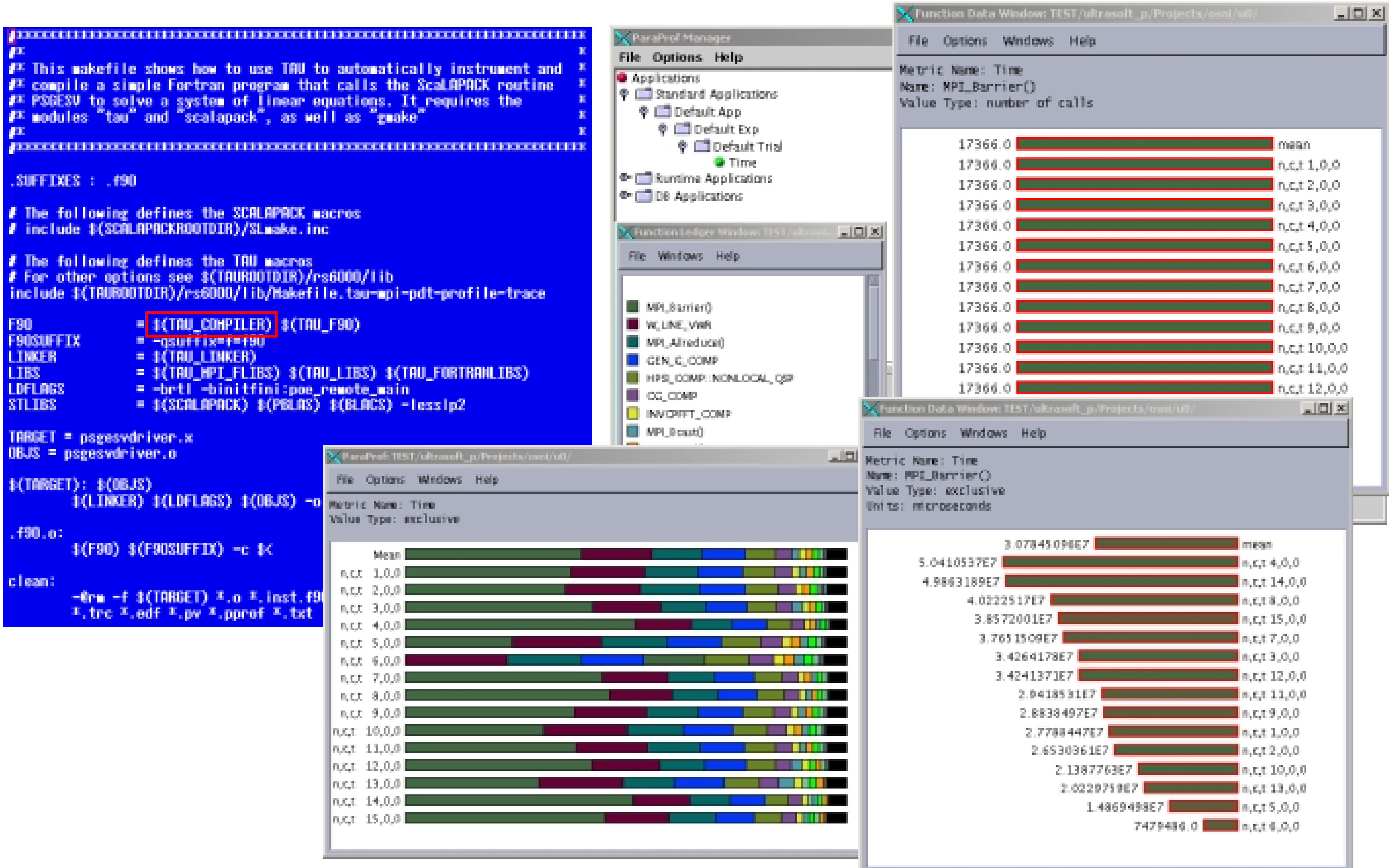


OPT++ *is used in protein energy minimization problems (shown here is protein T162 from CASP5, courtesy of Meza , Oliva et al.)*



Omega3P is a parallel distributed-memory code intended for the modeling and analysis of accelerator cavities, which requires the solution of generalized eigenvalue problems. A parallel exact shift-invert eigensolver based on PARPACK and ***SuperLU*** *has allowed for the solution of a problem of order 7.5 million with 304 million nonzeros. Finding 10 eigenvalues requires about 2.5 hours on 24 processors of an IBM SP.*

Use of ACTS Tools



Outreach Efforts

Yearly Workshop at LBNL

- ***Building Robust Scientific and Engineering High-End Computing Applications*, August 23-26, 2005**
- Previous (six) workshops (and other events) attended by more than 300 participants from DOE labs, academia and industry
- Tutorials and hands-on (on NERSC computers) delivered by tool developers

- Accumulation of expertise and user feedback on the use of the software tools and scientific applications that benefited from them
- Gallery of applications: ***http://acts.nersc.gov/MatApps***
- Collaboration with computer vendors, early access to new architectures allowing for porting and benchmarking

Application	Computational Problem	Software Tools	Highlights
MADCAP	Matrix factorization and triangular solves	ScaLAPACK	• 50% peak performance on an IBM SP • Nearly perfect scalability on 1024, 2048, 3072 and 4096 processors • Fast implementation of numerical algorithms
3-Charged Particles	Solution of large, complex unsymmetric linear systems	SuperLU	• Solves systems of equations of order 8.4 million on 64 processors in 1 hour of wall clock time • 30 GFLOPs
NWChem	Distribute large data arrays, collective operations	Global Arrays and LAPACK	• Very good scaling for large problems

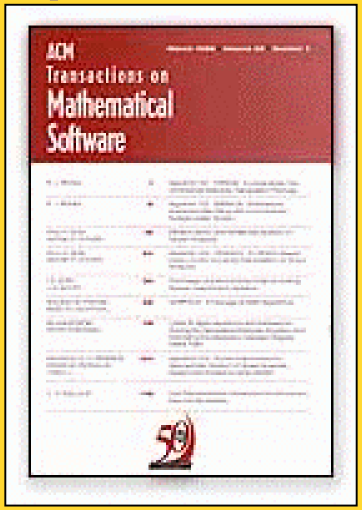
Recent Short Courses and Minisimposia

- ***An Introduction to Robust and High Performance Software Libraries for Solving Common Problems in Computational Sciences***, invited tutorial at the VECPAR'06, Rio de Janeiro, Brazil, July 2006.
- ***Short Course on the ACTS Collection***, SIAM CSE05 Conference, Orlando, FL.
- ***Numerical Software for Solving Problems in Computational Science and Engineering***, MS48, SIAM CSE05 Conference, Orlando, FL.

- *Enabling sciences and discoveries*
- *with*
- *high performance and scalability*

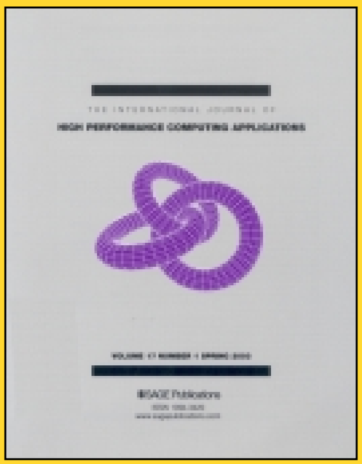
Journals Featuring ACTS Tools

September 2005 Issue

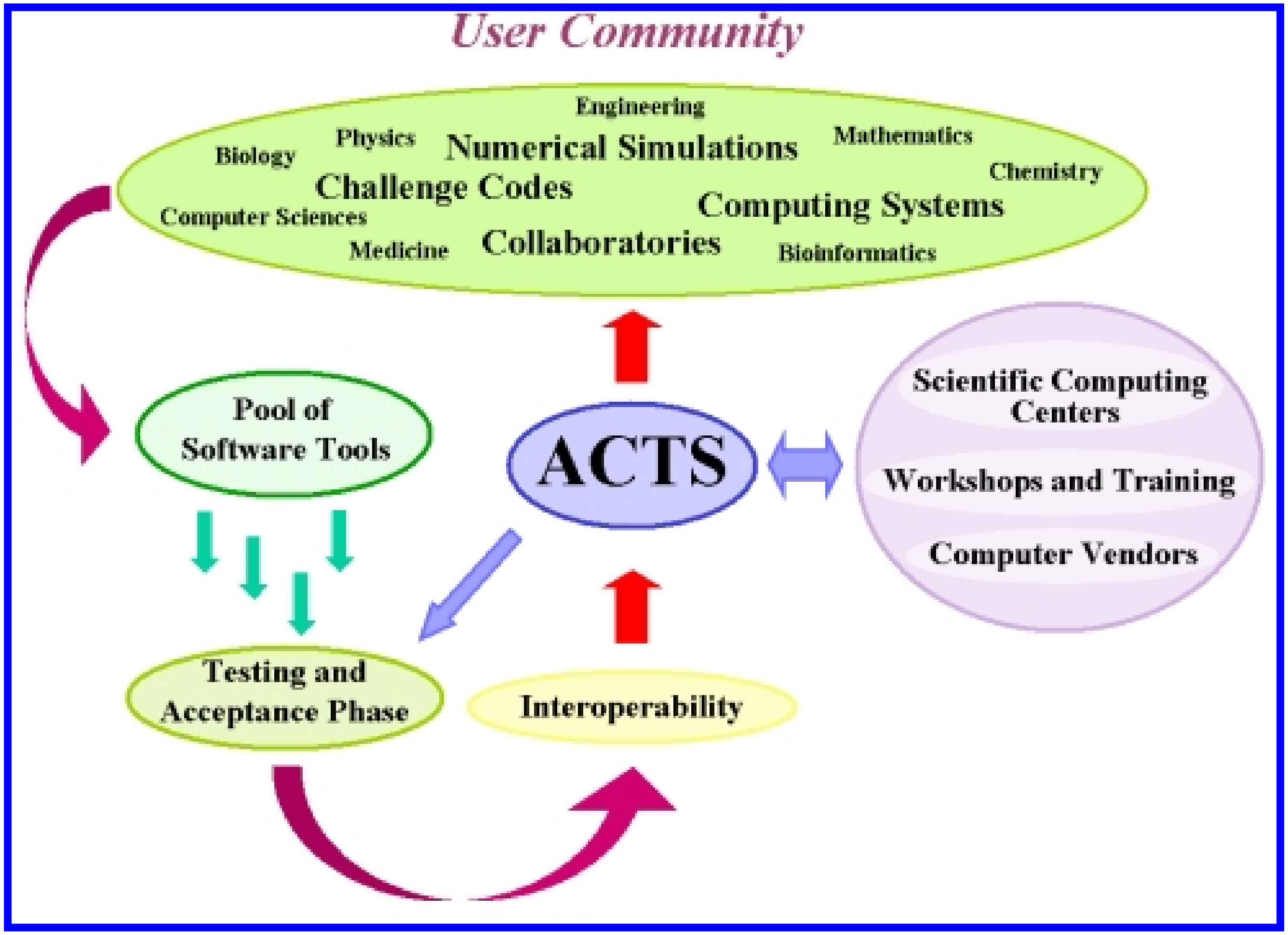


- *An Overview of the Advanced CompuTational Software (ACTS) Collection*, by T. Drummond and O. Marques
- *SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers*, by A. Hindmarsh, P. Brown, K. Grant, S. Lee, R. Serban, D. Shumaker and C. Woodward.
- *An Overview of SuperLU: Algorithms, Implementation, and User Interface*, by X. Li.
- *SLEPc: A Scalable and Flexible Toolkit for the Solution of Eigenvalue Problems*, by V. Hernandez, J. Roman and V. Vidal.
- *An Overview of the Trilinos Project*, by M. Heroux, R. Bartlett, V. Howle, R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring, A. Williams and K. Stanley.
- *Pursuing Scalability for hypre's Conceptual Interfaces*, by R. Falgout, J. Jones and U. Yang.

Summer 2006 Issue



- *A Component Architecture for High-Performance Scientific Computing*, by D. Bernholdt, B. Allan, R. Armstrong, F. Bertrand, K. Chiu, T. Dahlgreen, K. Damevski, W. Elwasif, T. Epperly, M. Govindaraju, D. Saltz, J. Kohl, M. Krishnan, G. Kumfert, J. Larson, S. Lefantzi, M. Lewis, A. Malony, L. McInnes, J. Nieplocha, B. Norris, S. Parker, J. Ray, S. Shende, T. Windus and S. Zhou.
- *CUMULVS: Interacting with High-Performance Scientific Simulations, for Visualization, Steering and Fault Tolerance*, by J. Kohl, T. Wilde and D. Bernholdt.
- *Advances, Applications and Performance of the Global Arrays Shared Memory Programming Toolkit*, by J. Nieplocha, B. Palmer, V. Tipparaju, M. Krishnan, H. Trease and E. Aprà.
- *The TAU Parallel Performance System*, by S. Shende and A. Malony.
- *High Performance Remote Memory Access Communication: The ARMCI Approach*, by J. Nieplocha, V. Tipparaju, M. Krishnan and D. Panda.



Have a nice Workshop!